ANALYSIS

30(127), April 1, 2015

ISSN 2278–5469 EISSN 2278–5450

Discovery

Optimized super pixel segmentation for natural images

Kabaleeswari T[1], Brinda T[2]

1.Dept of Computer Science and Engineering, Regional Centre of Anna University, Tirunelveli, Tamilnadu, India
2.Assistant Professor, Department of computer science, Regional Centre of Anna University, Tirunelveli, Tamilnadu, India

Publication History
Received: 05 February 2015
Accepted: 08 March 2015
Published: 1 April 2015

Citation
Kabaleeswari T, Brinda T. Optimized super pixel segmentation for natural images. *Discovery*, 2015, 30(127), 209- 215

ABSTRACT

Superpixel Segmentation using Lazy Random Walk Algorithm to attain the initial super pixel by selecting the seed location on input image according to the probabilities of each pixel and then boundaries of initial superpixels are attained according to the probabilities and commute time. The original superpixels are iteratively optimized with novel energy utility, which is distinct on the commute time and texture quantity. In proposed lazy random walk algorithm the performance of superpixel is superior with relocating the midpoint location of superpixel and isolating the large into miniature ones. LRW algorithm with self-loops has the qualities of segmenting the frail boundaries and difficult texture regions extremely fit with the new global probability maps and the commute time approach. The experimental results have confirmed that proposed method achieves recovered act than preceding super pixel approaches.

Index Terms: Lazy random walk, commute time, optimization, superpixel, texture

1. INTRODUCTION

A Superpixel can be described as a cluster of pixels which have similar uniqueness. Superpixels can be especially useful for image segmentation. There are a lot of algorithms available to segment superpixels. Superpixels are bigger related region of pixel and sometime also called image patches. Superpixels are generally distinct as constricting and combining homogenous pixels in the image, which have been extensively used in many

computer vision applications like image segmentation and object recognition. The superpixel concept as essential the perceptually homogenous regions using the normalized cuts (NCuts) algorithm.

Superpixels are often used as a pre-processing step for superior level algorithms, that fall the computations from millions of pixels to thousands of superpixels .Superpixel can cause substantial speed-up of subsequent processing since the number of superpixels of an image varies from 25 to 2500, in compare to hundreds of thousands of pixels. Super pixels are the consequence of perceptual grouping of pixels, or seen the former method approximately, the results of an image over segmentation. Superpixels carry additional information than pixels and align enhanced with image edges than rectangular image patches.

The pixel-grid representation is an "artifact" of a digital imaging process and not a usual one. Most of the image processing algorithms visualise image with the use of pixel-grid, as the basic version. Various stochastic models of images are often defined on this regular grid. It would be more usual, and most probably more efficient, to work with meaningful entities obtained from a low-level grouping process. The consequence of over-segmentation partitions the image into fewer numbers of segments known as superpixels. A Superpixel can be described as a cluster of pixels which have similar uniqueness. Superpixels can be especially useful for image segmentation. There are a lot of algorithms available to segment superpixels. Superpixels are bigger related region of pixel and sometime also called image patches.

Super pixels are the consequence of perceptual grouping of pixels, or seen the former method approximately, the results of an image over segmentation. Superpixels carry additional information than pixels and align enhanced with image edges than rectangular image patches. The main merit of superpixel is to offer a more usual and perceptually significant illustration of the input image. Therefore, comparison is done by conventional pixel illustration of the image. The superpixel description deeply decreases the amount of image primitives and progresses the delegate effectiveness and it is more suitable and successful to calculate the region based visual features by superpixels, which will give the vital benefits for the vision tasks like object recognition.

Every different superpixel generation approach has their own advantages and disadvantages that may enhance serve the need for processing the respective image. For example, if adherence to image boundaries is of paramount importance, the graph-based method may be a perfect selection. Conversely, if superpixels are to be used to construct a graph, a method that creates a more normal lattice is probably a better choice.

Since, no one ideal approach exists which constitutes for all applications, the following properties are generally desirable for superpixels: 1) Superpixels should exhibit good adherence to image boundaries. 2) At pre-processing step, superpixels should have reduced computational complexity, should be fast to compute, memory efficient, and simple to use. 3) At actual segmentation step, superpixels are supposed to linearly enlarge the speed and improve the quality of the consequences as well.

The main advantage of superpixel is to provide a more usual and perceptually significant illustration of the input image. Consequently, compared to the traditional pixel representation of the image, the superpixel representation very much reduces the number of image primitives and improves the representative efficiency. There is a large amount of literature on automatic superpixel algorithms, for example, normalized cuts [4], mean shift algorithm [6], graph-based method [2], Turbopixels [3], SLIC superpixels [13] and optimization-based superpixels [5]. conversely, each superpixel method has its own advantage and drawback that may be better suited for a particular application.

It is still challenging to develop a high quality superpixel algorithm, which avoids the under-segmentation and locally groups the pixels respecting the intensity boundaries. The desired properties of an ideal superpixel algorithm should not only adhere well to object boundaries of image, but also maintain the compact constrains in the complicated texture regions. In order to satisfy these desired requirements, we develop a new image superpixel segmentation method by the lazy random walk (LRW) and energy optimization algorithm to achieve better performance than the previous approaches.

Figure 1 Superpixel Image

Figure1 shows the example of image segmented using Achanta et al, SLIC algorithm [13] into superpixels of approximate size 64, 256 and 1024 pixels. The superpixels are compact, uniform in size, and adhere well to region boundaries.

2. RELATED WORK

The existing superpixel approaches can be roughly classified into two categories. The first category of algorithms that do not consider the compactness constraints during the superpixels generation procedure, such as mean shift, and graph based

algorithms. These algorithms produce superpixels by over-segmenting the image, in order to avoid the superpixels crossing the object boundaries. Non consideration of compactness constrains produces the superpixels of highly irregular shapes and sizes. The second category of superpixel algorithms considers the compactness constrains, such as normalized cuts, lattice cut, TurboPixels, and graph cut approaches.

M. Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa [1] has proposed an objective utility consists of two mechanisms: entropy rate of a random walk on a graph and a balancing term. This method starts with partition the input image into K superpixel and then graph is constructed. The entropy rate helps to compact and homogeneous clusters cheering separation of images on perceptual boundaries and helping superpixels overlapping with merely a particular object.

Fezenszwalb and Huttenlocher [2] has proposed graph-based segmentation scheme that has been used to generate superpixels. That algorithm performs an agglomerative clustering of pixel nodes on a graph, such that each segment, or superpixel, is the shortest spanning tree of the constituent pixels. It has been used for depth estimation. It wasO(NlogN) complex and was quite fast in practice as compared to Normalized cut.

A. Levinshtein, A. Stere, K. Kutulakos, D. Fleet, S. Dickinson, and K. Siddiqi [3] has proposed TurboPixel algorithm employing the level set based geometric flow evolution from the homogenously located seeds in the image. Theoretically that algorithm places initial seeds in location, iterate the image with additional steps until no additional progression is possible. Iteration consists of developing the boundary for T time steps, manipulative the skeleton of unassigned region, moderize the speed of each boundary pixel and unassigned pixel in boundary's vicinity.

Ren and Malik [4] propose using Normalized Cut (NCut) forsuperpixel segmentation. NCut has the nice property of producing superpixels with similar sizes and compact shapes. One drawback of NCut is its computational requirement— it takes several minutes for segmenting an image of moderate (481x321) size.

O. Veksler, Y. Boykov, and P. Mehrani [5] has proposed the superpixel separating difficulty with in an energy decrease skeleton and also optimize through graph cuts. It showed a dissimilarity of the vital energy, which permits a transaction between a fewer usual tessellation but extra precise boundaries or superior efficiency. A superpixel is an image space which is better aligned with intensity edges than a rectangular patch.

D.Comaniciu and P.Meer [6] has proposed mean shift algorithm that generates image segments by recursively moving to the kernel smoothed centroid for every data point in the pixel feature space, effectively performing a gradient ascent .The generated segments/superpixels can be large or small based on the input kernel parameters, but there is no direct control over the number, size, or compactness of the resulting superpixels.

A superpixel lattice was generated by [7] by finding optimal vertical (horizontal) seams/paths that cut the image, within vertical (horizontal) strips of pixels, using graph cuts on strips of the image. While superpixel lattice allows control of the size, number, and compactness of the superpixels, the quality and speed of the output strongly depend on pre-computed boundary maps.

Lattice superpixel [8] was derived from global optimization. The superpixel generation is initialized with a grid, and the graph cut algorithm is adopted to iteratively optimize the vertical and horizontal seams.

L.Grady [9] discussed about the interactive image segmentation. It used the prelabeled pixels, which was used to avoid the boundary crossing. The unlabeled pixels are present in the top of the list in the random walker that establishes the probability in random walker. Actually it fully depends for the associatation with the current seed and all its equivalent seed. But it ignores the dependency among the seeds, not only with the equivalent seeds. It can't able to work for frail boundary images.

3. LRW APPROACH

The RW algorithm has been used extensively for interactive image segmentation in the image processing and computer vision literatures [12]. The RW algorithm calculates the first arrival probability to facilitate a random walk starts at one pixel first reaches one of the seeds with each label, and then that pixel is denoted as the same label with maximum probability of the equivalent seed. A random walk starts from a pixel must first arrive at the position of the prelabeled seed, and thus it only considers the local relationship between the current pixel and its corresponding seed. This first arrival probability ignores the whole relationship between the current pixel and other seeds. As denoted by Grady [12], these limitations of the original RW method give the reason that it suffers from the weak boundary and complex texture segmentations.

In order to make full use of the global relationship between the pixel and all the seeds, we add the self-loop over the graph vertex to make the RW process lazy, which is inspired by the original LRW concept [12]. However, the original LRW was initially proposed for the website data classifying and mining applications.

4. SUPERPIXEL INITIALIZATION

The superpixel initialization is computed by commute time. CT_{ij} to represent the predictable numbers of steps used for a lazy random walk that begins at node v_i to reach node $v_j$and then return to v_i. CT_{ij} is called the commute time. Commute time is

computed from the Laplacian spectrum using the discrete function. The commute time for image segmentation using the eigenvector corresponding to the smallest eigenvalue of the commute time matrix. The Eigen vector does not change in the direction in the transformation. The value of change in length of the vector is known as Eigen value.

A. Weight estimation

The edge weights are calculated on dissimilarity between pixel intensities .First a graph G=(V,E) is constructed on given input image $I(x_i)$. In that graph includes set of nodes V that match to a pixel in the image and edges in E connect certain pair of neighboring pixels. The degree of each vertex is computed as

$$d_i = \sum_j W_{ij}$$

The degree vertex means to be sum of the weights of all the edges that occurrence on v_i. The edge-weight calculation process is used to characterize the image intensity changes. This edge-weight evaluates the relationship between two neighboring nodes $v_i$and v_j ,and therefore $W_{ij}$is described by the Gaussian weighting function given in eqn1

$$w_{ij}=\exp\left(-\frac{\|g_i-g_j\|^2}{2\sigma^2}\right) \qquad (1)$$

where g_i and g_j denote the image intensity values at two nodes,v_i and v_j, and 6 is identity factor. The value of 26^2 is 1/30.

B. Adjacency Matrix Estimation

The neighborhood relations are summarized by the adjacency matrix. The adjacent node shares the common vertex and it is given eqn2.

$$W_{ij}=\begin{cases} 1-\alpha if i=j, \\ \alpha. w_{ij} if i\sim j, \\ 0 \quad otherwise, \end{cases} \qquad (2)$$

Where $i\sim j$ means node $v_i$and node $v_j$are the closest nodes. Node is nothing but the pixel in an image.

Later than row normalize the adjacency matrix it is to identify the alternation probability matrix is given in eqn 3

$$P_{ij}=\begin{cases} 1-\alpha if i=j, \\ \alpha.\frac{w_{ij}}{d_i} if i\sim j, \\ 0 \quad otherwise \end{cases} \qquad (3)$$

Eqn3 can also be write down as $P=(1-\alpha)I+\alpha D^{-1}W$ where D is a diagonal matrix with the (i,i)-th entry having the value d_i. This means with the probabilityα, it follows one link which connects the vertex of the current position and is chosen with the probability proportional to the weight of the link, and with the probability (1-α) , just saying at the current position.

C. Laplacian Matrix Estimation

The graph laplacian directly incorporates a graph structure describing the local neighbourhood relation between data points. Laplacians finds a smooth function such that it's values in seed pixels are close to the associated labels and it is allowed to vary only on low-density regions of the input space. The neighbourhood relations are summarized by the adjacency matrix. The laplacian matrix is given in eqn4

$$L_{ij}=\begin{cases} d_i if i=j \\ -\alpha w_{ij} if i\sim j \\ 0 \quad otherwise \end{cases} \qquad (4)$$

Eqn4 can be also write-down as

$$L=D-\alpha W \qquad (5)$$

Here D is the diagonal matrix.W is the adjacency matrix.D_{ij} is the degree of the ith vertex v_i.

$CT_{ij}$to represent the predictable number of steps used for a lazy random walk that begins at node v_i to reach node $v_j$and then return to v_i.CTij is called the commute time between v_i and v_j.

After that obtain the Normalized laplacian matrix using the given eqn 6

$$L=(I-\alpha D^{-1/2}WD^{-1/2}) \qquad (6)$$

Later than normalizing the commute time toone that is the amount of commute time from a node to other nodes. Normalized CT is given in eqn 7

$$CT_{ij}=\begin{cases} 1-L_{ij}{}^{-1} if i\neq j \\ 1 \quad if i=j \end{cases} \qquad (7)$$

The commute time is inversely propotional to the probability. The LRW algorithm will be at a pixel x_i with the boundary likelihood probabilities of label l as given in eqn 8

$$f_l=(I-\alpha S)^{-1}y t(8)$$

Where I is the identity matrix and S is given by

$$S=D^{-1/2}WD^{-1/2}$$

Here D is the diagonal matrix and W is the adjacency matrix .

At last reach the labeled boundaries of super pixel from the commute time is given in eqn 9

$R(x_i)=argmin_{lk}CT(cl_k,x_i)=argmax_{lk}CT(cl_k,x_i)$(9)

clk stands for the center of the l-th super pixel, and the label lk*is* allocated to each pixel $x_i$to find the boundaries of super pixels. Next the label of the seed among the smallest amount commute time is allocated to the related pixel the same as the concluding superpixel label.

LRW Based Superpixel Initialization Algorithm

Input: Input image $I(x_i)$ and an integer of initial seeds K

Step1: Define an adjacency matrix.

Step2: Construct the laplacian matrix $S=D^{-1/2}WD^{-1/2}$ in which D is a diagonal matrix

Step3: Compute $f_{lk}=(I-\alpha S)^{-1}y_{lk}$

Step4: Compute $R(x_i)=argmax_{lk}CT(cl_k,x_i)$ to obtain the labels by assigning label $R(x_i)$ to each pixel x_i.

Step5: obtain superpixel S_{lk} based on k.

ouput: the initial superpixel results S_{lk}

5. SUPERPIXEL OPTIMIZATION

In this module the performance of super pixels is improved with energy optimization function by via the compactness constraints. The superpixel algorithm rely on minimizing objective utilities to implement color homogeneity. Optimization techniques are built in the lead progressively accumulating cuts, or grow superpixels starting from various estimated centers.

$$E=\sum_l(\mathrm{Area}(S_l)-\mathrm{Area}(\bar{S}))^2+\sum_l\widetilde{W_x}\,CT\left(c_l^{n},x\right)^2 \quad (10)$$

$CT\left(c_l^{n},x\right)-->$Commute time between seed point c_l^{n} and pixel x.

$\widetilde{W_x}$is a consequence task it is used to find the commute time between the seed and the pixel.

The minimization of smooth item gives the optimal relocation center position of superpixel. The new center relocates is given in eqn 11

$$C_l^n=\frac{\sum_l\widetilde{W_x}\frac{CT(c_l^{n-1},x)}{\|x-c_l^{n-1}\|}x}{\sum_l\widetilde{W_x}\frac{CT(c_l^{n-1},x)}{\|x-c_l^{n-1}\|}} \quad (11)$$

The texture characterize of local binary pattern (LBP) to calculate the texture information. The LBP value of each pixel is given in eqn 12

$$LBP_i^{q,r}=\sum_{t=0}^{q-1}s(g_t-g_i)2^t \quad (12)$$

Where *q* is the gray level of LBP, *r* is the radius of the circle around pixel *i*, and *g* means the gray value of image. *q* = 8 and *r* = 1. Based on the LBP texture feature, the area of superpixel is given in eqn 13

$$Area(Sl)=\sum_{i\in Sl}LBP_i,$$

$$Area(\bar{S})=\frac{\sum_{i\in Sl}LBP_i}{N_{sp}} \quad (13)$$

where$Area(Sl)$is the region of superpixels, and $Area(\bar{S})$ is the average region of the superpixel and Nspis the user described quantities of superpixels. The region of superpixel is huge when it includes greatly texture information. The parameter Th manages the amount of iterations in superpixel optimization process. The ratio of superpixel area and the average superpixel area is greater than threshold the large super pixel is separated into two little superpixels. Here threshold helps to the superpixel whether it is split or not. The Prinicipal component analysis method is used to split the superpixel. Later than splitting process, the 2 new superpixels are found and the equivalent two new centers $C_{lnew,}1$, $C_{lnew'}2$ is given in eqn 14

$$C_{lnew,}1=\frac{\sum_{\{(x-c_l).s>0\}}\widetilde{W_x}\frac{CT(c_l,x)}{\|x-c_l\|}x}{\sum_{\{(x-c_l).s>0\}}\widetilde{W_x}\frac{CT(c_l,x)}{\|x-c_l\|}} \quad (14)$$

$$C_{lnew'}2=\frac{\sum_{\{x|x\in S_l,(x-c_l).s<0\}}\widetilde{W}\frac{CT(c_l,x)}{\|x-c_l\|}x}{\sum_{\{x|x\in S_l,(x-c_l).s<0\}}\widetilde{W}\frac{CT(c_l,x)}{\|x-c_l\|}}$$

Superpixel Optimization Algorithm

Input: Initial Superpixel S_l and an integer N_{sp}

Step1: To Compute the center of the Superpixl S_l

Step2: Splitting the Large Superpixel into two small Superpixel and obtain two new center

Step3: Refine Initial Superpixel S_l with center of the Superpixel S_l, two new centers

Step4: Run steps 1 to 3 iteratively until convergence

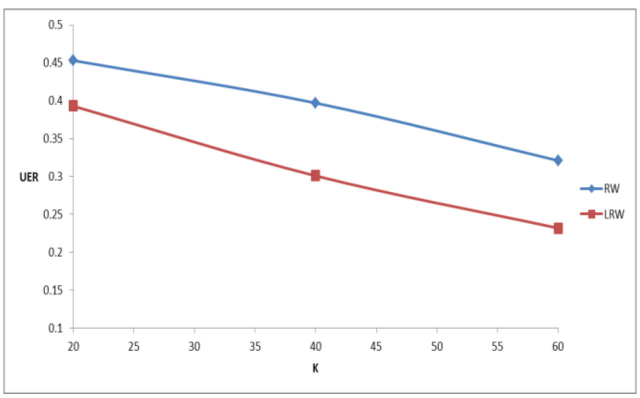
Output: The final optimized superpixel results

6. EXPERIMENTS

In this section, the superpixel results are analyzed to explain the influence with different parameter settings. The proposed superpixel algorithm is then qualitatively tested with several representative examples in BSD benchmark. In order to obtain an objective and intuitive comparison and then quantitatively evaluate this algorithm by three different error metric measurements. Comparison results between RW method and LRW algorithm before and after optimization, and analyze the computational complexity of the representative superpixel algorithms.

A. Under Segmentation Error (USE)

The under segmentation error estimation capacity checks the deducting area by the superpixel that overlies the given ground-truth segmentation. K is the seed points.



.

Figure 2 The Curve of Under Segmentation Error

This graph shows the performance of the LRW algorithm achieves better performance than RW algorithm**.**

B. Boundary Recall (BR)

Boundary recall measurement calculates the ratio of the ground truth boundaries that decrease within the adjacent superpixel boundaries.

Lazy Random Walk algorithm precisely calculates the boundary than Random Walk algorithm.

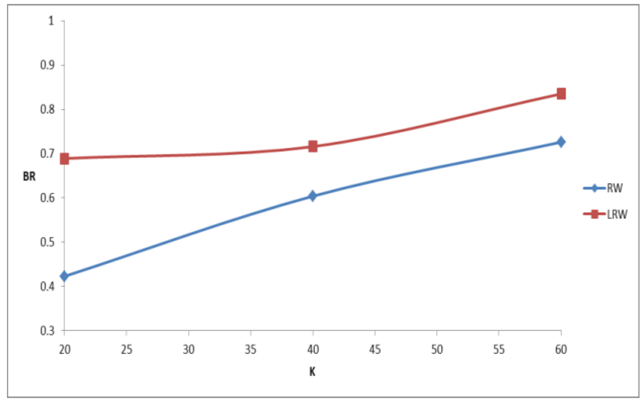


Figure 3 The Curve of Boundary Recall

C. Achievable Segmentation Accuracy (ASA)

Achievable Segmentation Accuracy calculates the maximum achievable accuracy by labeling every superpixel with the label of ground truth segmentation that has the major overlie area.

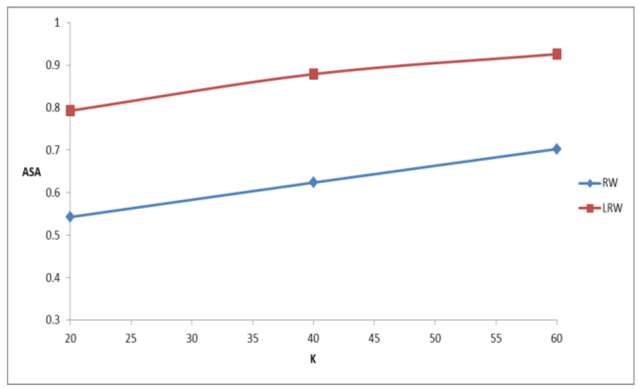


Figure 4 The Curve of Achievable Segmentation Accuracy

Lazy Random Walk algorithm achieves highest accuracy than Random Walk algorithm.

7. CONCLUSION

We have present a natural image superpixel approach using the Lazy Random Walk and energy optimization algorithm. This method first starts with the Lazy Random Walk algorithm to attain the initial superpixel consequences by selecting the seed locations on input image. Then it further optimize the labels of superpixels to improve the reliability and boundary devotion performance by relocating the center positions of superpixels and separating the huge superpixels into two small homogeneous ones in an energy optimization skeleton. The experimental results have established that the super pixel algorithm accomplishes enhanced performance than the preceding well-known superpixel approaches. This algorithm is capable of obtaining the good boundary adherence in the difficult texture and weak edge regions, and the proposed optimization strategy considerably improves the quality of

superpixels. In future work, optimized Superpixel segmentation can be obtained using various texture feature methods.

REFERENCE

1. M. Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, “Entropy rate superpixel segmentation,” in Proc. IEEE CVPR, Jun. 2011, pp. 2097–2104.
2. P. Felzenszwalb and D. Huttenlocher, “Efficient graph-based image segmentation,” Int. J. Comput.Vis., vol. 59, no. 2, pp. 167–181, 2004.
3. A. Levinshtein, A. Stere, K. Kutulakos, D. Fleet, S. Dickinson, and K. Siddiqi, “Turbo pixels: Fast super pixels using geometric flows,” IEEETrans. Pattern Anal. Mach. Intell., vol. 31, no. 12, pp. 2290–2297, Dec. 2009 .
4. X. Ren and J. Malik, “Learning a classification model for segmentation,” in Proc. 9th IEEE ICCV, Oct. 2003, pp. 10–17.
5. O. Veksler, Y. Boykov, and P. Mehrani, “Superpixels and supervoxels in an energy optimization framework,” in Proc. ECCV, 2010, pp. 211–224.
6. D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 5, pp. 603–619, May 2002.
7. A. Moore, S. Prince, J. Warrell, U. Mohammed, and G. Jones, “Superpixel lattices,” in Proc. IEEE CVPR, Jun. 2008, pp. 1–8.
8. A. Moore, S. Prince, and J. Warrel, “Lattice cut—Constructing superpixels using layer constraints,” in Proc. IEEE CVPR, Jun. 2010, pp. 2117–2124.
9. L. Grady, “Random walks for image segmentation,” IEEE Trans. Pattern Anal. Mach Intell., vol. 28, no. 11, pp. 1768–1783, Nov. 2006.
10. C. Couprie, L. Grady, L. Najman, and H. Talbot, “Power watershed: A unifying graph-based optimization framework,” IEEE Trans. PatternAnal. Mach. Intell., vol. 33, no. 7, pp. 1384–1399, Jul. 2011.
11. D. Martin, C. Fowlkes, D. Tal, and J. Malik, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,” in Proc. 8th IEEE ICCV, Vancouver, BC, Canada, Jul. 2001, pp. 416–423.
12. D. Zhou and B. Sch¨olkopf, “Learning from labeled and unlabeled data using random walks,” in Proc. DAGM, 2004, pp. 237–244.
13. R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fsua, and S. Süsstrunk, “SLIC superpixels,” EPFL, Lausanne, Switzerland, Tech. Rep. 149300, 2010.